(12) UK Patent Application (19) GB (11) 2 111 805 A

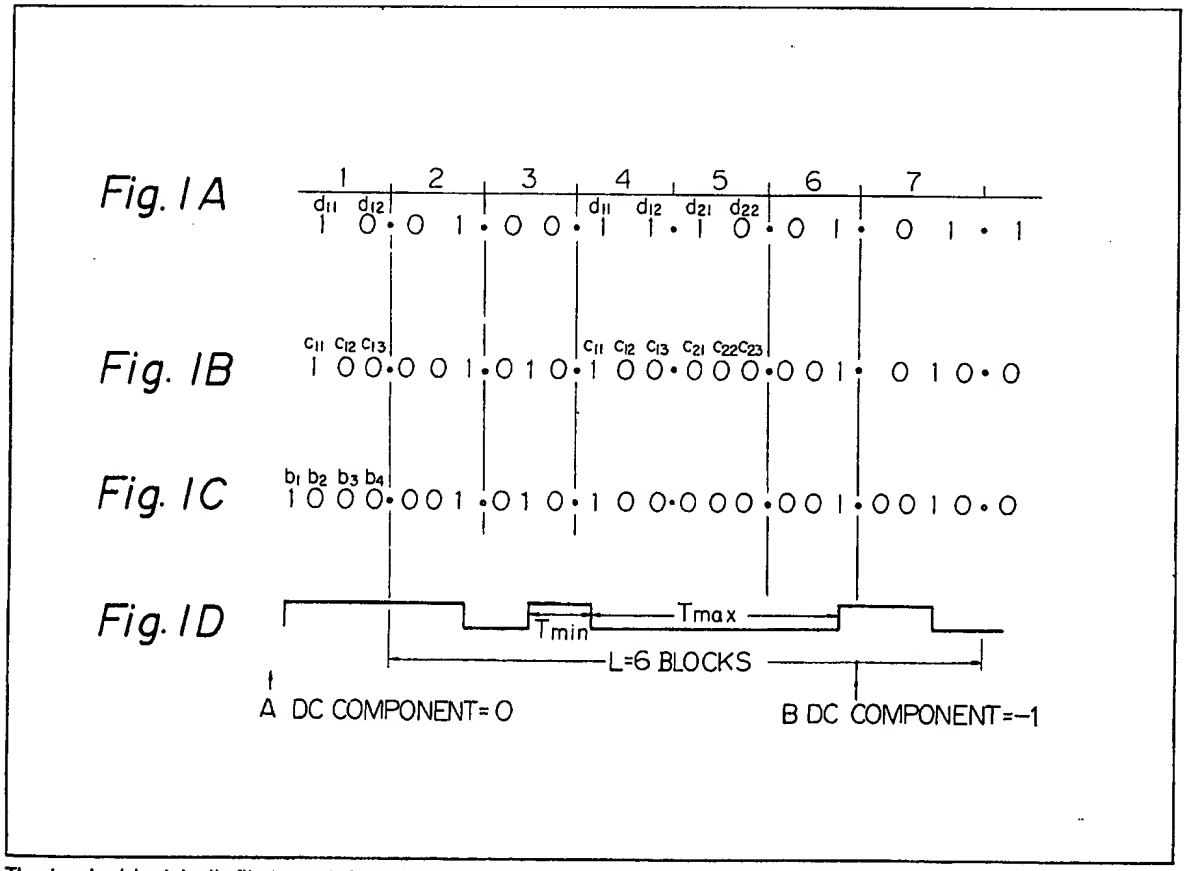
(21) Application No 8230737
(22) Date of filing 27 Oct 1982
(30) Priority data
(31) 56/173258
(32) 29 Oct 1981
(33) Japan (JP)
(43) Application published 6 Jul 1983
(51) INT CL³ H04L 25/49
(52) Domestic classification H4P DD
(56) Documents cited GB A 2049371 GB 1590404 GB 1498819
(58) Field of search H4P
(71) Applicants Pioneer Electronic Corporation, (Japan), 4-1 Meguro 1-chome, Meguro-ku, Tokyo 153, Japan.

(72) Inventors Yoshiaki Moriyama, Kenji Yamagata.

(74) Agent and/or Address for Service Marks and Clerk, 57-60 Lincoln's Inn Fields, London WC2A 3LS.

# (54) Method for converting a binary data train

(57) In order to obtain sufficiently long minimum interval between transitions $T_{min}$ and a wide detection window $T_w$ of data conversion, and to reduce the dc or the low frequency component, a method for converting a binary data train comprises a first step for dividing the binary data train into a plurality of successive blocks each having M (M being a natural number) bit data, a second step for converting the M bit data into N (N being a natural number, and where $N \geqq M+1$), and a third step for converting the N bit data in every L (L being a natural number) blocks into J (J being a natural number) bit data.




GB 2 111 805 A

The drawing(s) originally filed were informal and the print here reproduced is taken from a later filed formal copy.

Fig. 1A

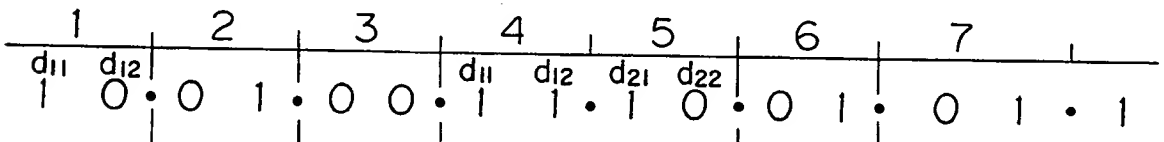



Fig. 1B

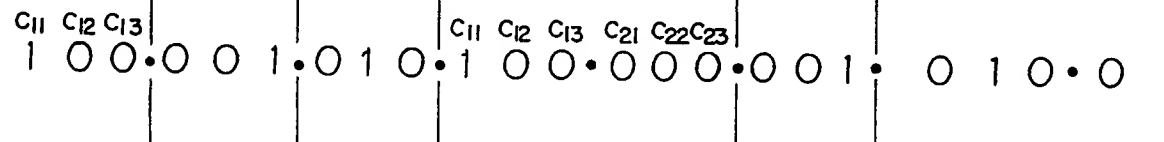



Fig. 1C

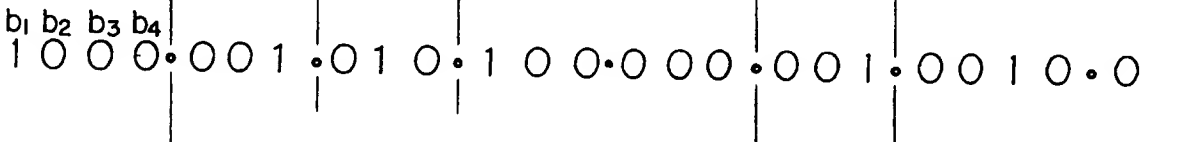
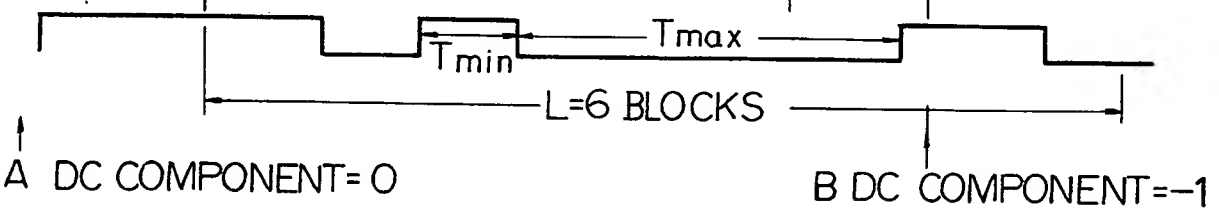



Fig. 1D



# Fig. 2

| $d_{11}$ $d_{12}$ | $c_{11}$ $c_{12}$ $c_{13}$ |
|---|---|
| 0 0 | 0 1 0 |
| 0 1 | 0 0 1 |
| 1 0 | 1 0 0 |
| 1 1 | 1 0 1 |

do Fig 2
then if there is a 11 then
apply Fig 3

# Fig. 3

| $d_{11}$ $d_{12}$ $d_{21}$ $d_{22}$ | $c_{11}$ $c_{12}$ $c_{13}$ $c_{21}$ $c_{22}$ $c_{23}$ |
|---|---|
| 0 1 • 1 0 | 0 1 0 • 0 0 0 |
| 0 1 • 1 1 | 0 0 1 • 0 0 0 |
| 1 1 • 1 0 | 1 0 0 • 0 0 0 |
| 1 1 • 1 1 | 1 0 1 • 0 0 0 |

m, n, d, k = 2, 3, 1, 7

apply first conversion of Fig's 2 & 3 then apply conversion of Fig 4 or Fig 5

# Fig. 4

| $c_{i1}$ $c_{i2}$ $c_{i3}$ | | $b_1$ $b_2$ $b_3$ $b_4$ |
|---|---|---|
| 0 1 0 | $P_1$ | 0 0 1 0 |
| | $P_2$ | 0 0 0 0 |
| 0 0 1 | $P_1$ | 0 1 0 1 |
| | $P_2$ | 0 0 0 1 |
| 1 0 0 | $P_1$ | 1 0 1 0 |
| | $P_2$ | 1 0 0 0 |
| 0 0 0 | $P_1$ | 0 1 0 0 |
| 1 0 1 | $P_1$ | 1 0 0 1 |

Select between $P_1$ & $P_2$ to control DC and K

Fig. 5

| $c_{i1}$ $c_{i2}$ $c_{i3}$ | | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ |
|---|---|---|---|---|---|---|
| 0 1 0 | $P_1$ | 0 | 1 | 0 | 1 | 0 |
| | $P_2$ | 0 | 0 | 0 | 1 | 0 |
| | $P_3$ | 0 | 1 | 0 | 0 | 0 |
| 0 0 1 | $P_1$ | 0 | 0 | 1 | 0 | 1 |
| | $P_2$ | 0 | 1 | 0 | 0 | 1 |
| | $P_3$ | 0 | 0 | 0 | 0 | 1 |
| 1 0 0 | $P_1$ | 1 | 0 | 1 | 0 | 0 |
| | $P_2$ | 1 | 0 | 0 | 1 | 0 |
| | $P_3$ | 1 | 0 | 0 | 0 | 0 |
| 0 0 0 | $P_1$ | 0 | 0 | 1 | 0 | 0 |
| | $P_2$ | 0 | 0 | 0 | 0 | 0 |
| 1 0 1 | $P_1$ | 1 | 0 | 1 | 0 | 1 |
| | $P_2$ | 1 | 0 | 0 | 0 | 1 |

m, n, d, K = 3, 5, 0,

select $P_1$ ~ $P_3$ to control = DC & K

2111805

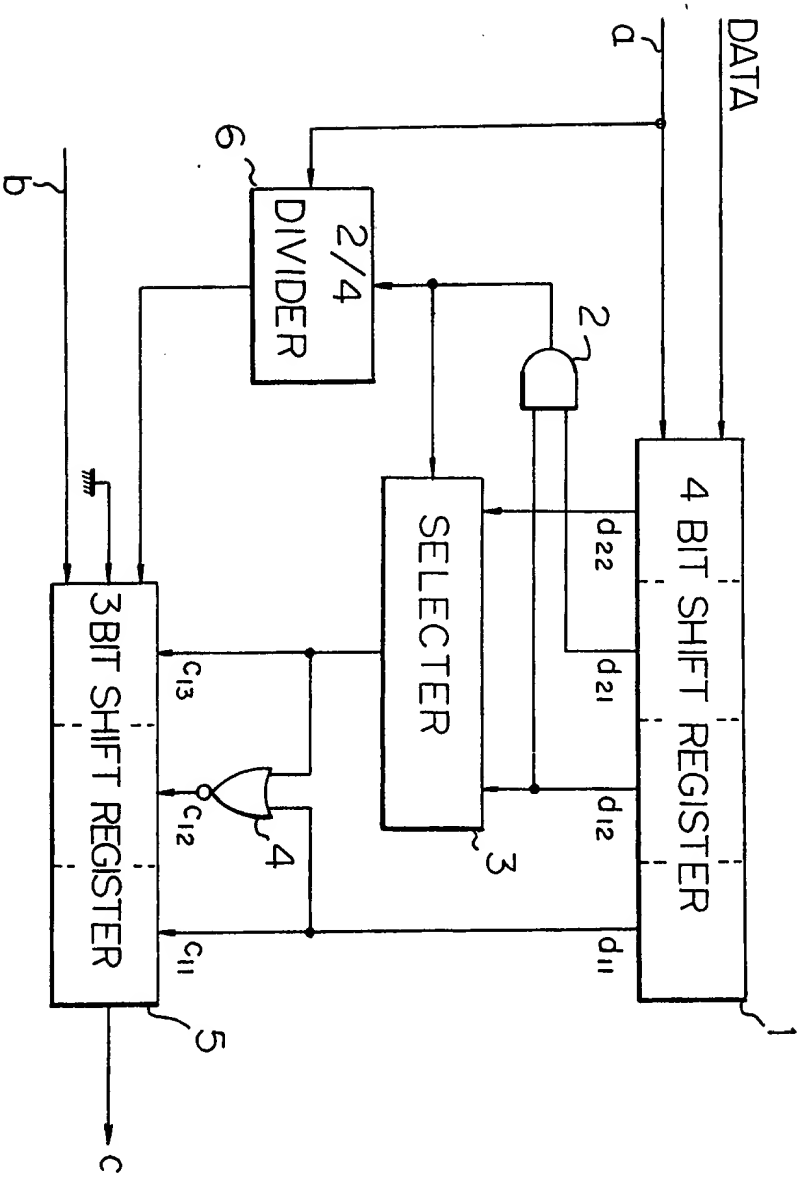



Fig. 6A

2111805

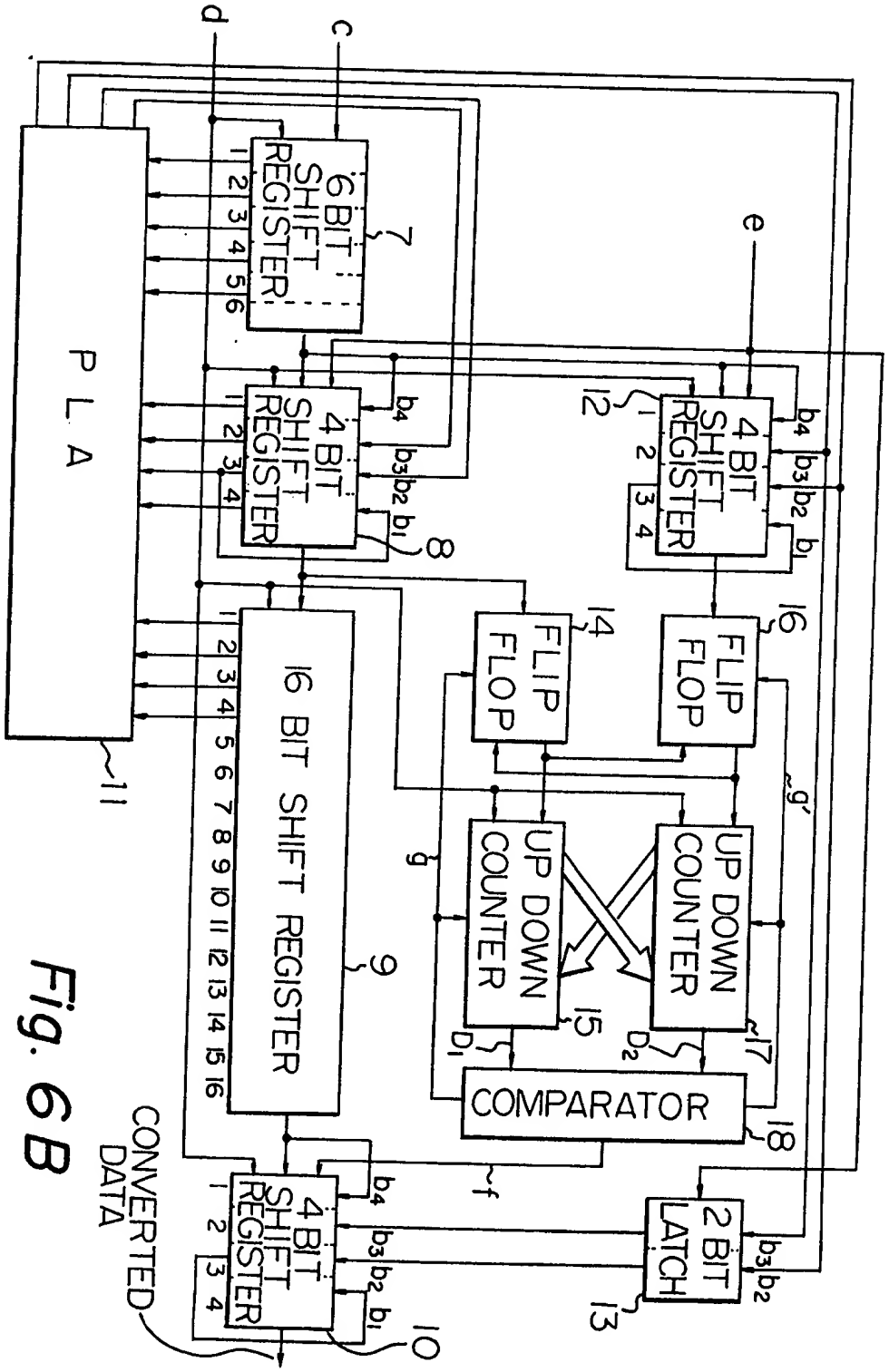



Fig. 6B

2111805

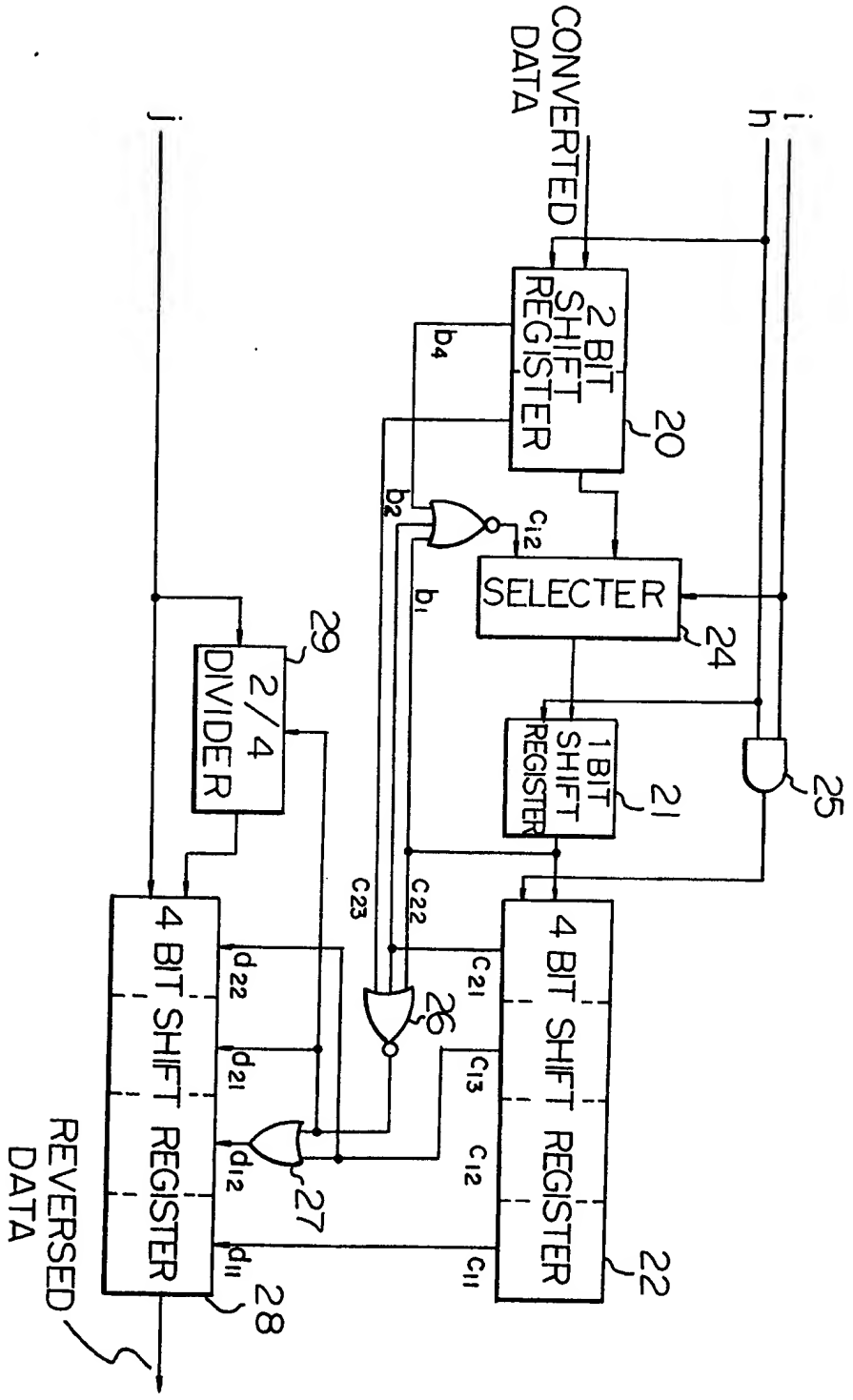



Fig. 7

SPECIFICATION

**Method for converting a binary data train**

*Field of the invention*

The present invention relates to a method for converting a binary data train, and more specifically to a binary data conversion method which is to be applied in such an occasion that binary data is recorded in a high-density recording medium, or that binary data is transmitted through a transmission system having a relatively narrow bandwidth.

*Description of the prior art*

Various methods of binary data modulation are proposed and put to practical use in order to record data onto a high-density recording medium such as a magnetic tape, a magnetic disc, or an optical disc, or to transmit binary data through a transmission system.

Such methods of binary data modulation, especially for the high density recording or the high density transmission, can be treated as combination of a process of binary data conversion and a process of modulation of a binary signal train which is obtained by the binary data conversion. In such a binary data conversion process, a binary data train is divided into a series of blocks, each having M bit binary code (M is a natural number), and then, such an M bit binary code is converted into an N bit binary code. The process of modulation is generally either one of fundamental modulation processes of the NRZ1 (Non-Return to Zero Inverse) or the NRZ (Non-Return to Zero) type.

Generally, the process of modulation for the high density of recording must comply with the following conditions:

(i) The minimum interval (referred to as $T_{min}$ hereinafter) of the inversion of the recording signal is sufficiently long and the maximum interval (referred to as $T_{max}$ hereinafter) thereof is sufficiently short;

(ii) The window of detection (referred to as $T_w$ hereinafter), which is a time duration for detecting the recorded bits from a signal reproduced from the recording medium, is sufficiently wide; and

(iii) The signal to be recorded on the recording medium, which is obtained after modulation, does not contain a dc component or a low frequency component.

A long $T_{min}$ is suitable for reducing interference between two adjacent inversions, which enables a high density recording. On the other hand, a short $T_{max}$ is advantageous for self-synchronization.

Since the detection of the recorded bits is effected by detecting peak values of a reproduced signal wave in the case of magnetic recording system, a wide $T_w$, which determines the tolerance of the error of the position of detection, is advantageous for the high density recording. Also, in the case of a recording system employing a laser beam, the wide $T_w$ is suited for improving the signal to noise ratio due to an increased amplitude of the signal at the point of detection.

Furthermore, if a recording signal having a dc component or a low frequency component is applied to equipment including a transmission system which does not transmit the low frequency component, the waveform of such a signal would be distorted. In addition, such a dc component or a low frequency component would cause a malfunction of the servo systems of a recording system employing a laser beam. If the signal doesnot have a dc or a low frequency component as an essential part, a high pass filter can be employed to suppress low frequency noise or drift components.

Among the prior art methods of modulation, the Zero Modulation (ZM) and the Modified Miller ($M^2$) techniques satisfy the above-described condition (iii); however, in those methods of modulation, the time length of $T_{min}$ is almost the same as the time length T of a bit of the data before modulation ($T_{min} \fallingdotseq T$ in the case of the ZM, and $T_{min} = T$ in the case of $M^2$), and therefore, not sufficiently long.

A long $T_{min}$ ($T_{min} = 1.5T$) is obtained in the Three Position Modulation (3M) technique, however, it does not satisfy the above condition (iii).

The Eight to Fourteen Modulation (EFM) technique which has been proposed recently, has a long $T_{min}$ of 1.41T, and satisfies the condition (iii); however, it has the drawback of short $T_w$ of 0.47T. Furthermore, the EFM technique has a problem that the structures or the modulator and the demodulator tend to be complicated since 8 bit data is converted to 14 bit data which has a relatively large sized unit.

*Summary of the invention*

An object of the invention is therefore to provide a method for converting binary data in which a modulated signal does not have any dc or low frequency components and at least one of the $T_{min}$ and the $T_w$ thereof is sufficiently long.

Another object of the invention is to provide a converting method in which the structure of the demodulator can be simplified.

According to the present invention, a binary data conversion method comprises a first step of dividing a binary data train into a series of successive blocks, each having M bit data (M being a natural number), a second step of converting each of the M bit data into N (N being a natural number and equal to or larger than M + 1) bit data, and a third step for converting an N bit data in every L number of the blocks into J bit data.

The foregoing and other objects and advantages of the present invention will become more clearly understood upon review of the following description taken in conjunction with the accompanying drawings.

*Brief description of the drawings*

*Figures 1A* through *1D* are diagrams showing the relation between the steps of the data converting method according to the present invention;

*Figures 2* through *5* are graphs showing the transformations in each step of the data conversion;

*Figures 6A* and *6B* are circuit diagrams of an example of a conversion circuit according to the present invention; and

*Figure 7* is a circuit diagram of an example of a circuit for reversing the data which is converted by the circuit shown in Figures 6A and 6B.

*Detailed description of preferred embodiments*

Reference is first made to Figures 1A through 1D in which an embodiment of the data conversion method according to the present invention is depicted.

Figure 1A shows successive binary data which is to be recorded in such recording medium as a digital audio disc (DAD). The binary data is divided into a plurality of blocks each of which has two bits of data, and the groups are designated from No. 1 through No. 7 in the Figure.

Each block of 2 bit data in the blocks shown in Figure 1A is then converted to 3 bit data shown in Figure 1B according to rules shown in Figures 2 and 3 which will be explained hereafter.

More specifically, each 2 bit data block such as $d_{11}$ and $d_{12}$ shown in Figure 1A are in turn converted to 3 bit data $C_{11}$, $C_{12}$, and $C_{13}$ shown in Figure 1B in accordance with the rule shown in the truth table of Figure 2, which is expressed by equations as follows:

$$C_1 = d_1,\ C_3 = d_2,\ \text{and}\ C_2 = \overline{C_1 + C_3},$$

in which $d_1$ and $d_2$ are respectively higher bit and lower bit binary code of the 2 bit data, $C_1$ through $C_3$ are respective three binary codes of the 3 bit data, in the order of higher bit, middle bit, and lower bit.

However, if adjacent binary codes which belong to different blocks, such as $d_{12}$ and $d_{21}$ of blocks No. 4 and No. 5 shown in Figure 1A are 1, the conversion is effected in accordance with the rule shown in the truth table of Figure 3, which is expressed as follows;

$$C_{11} = d_{11},\ C_{13} = d_{22},\ C_{12} = \overline{C_{11} + C_{13}},$$

and

$$C_{21} = C_{22} = C_{23} = \overline{d_{12}} = \overline{d_{21}} = 0,$$

in which $d_{11}$, $d_{12}$, $d_{21}$, and $d_{22}$ are binary codes of 2 bit data in two successive blocks, and $C_{11}$, $C_{12}$, $C_{13}$, $C_{21}$, $C_{22}$, and $C_{23}$ are binary codes of 3 bit data after the conversion.

Furthermore, if adjacent binary bits belonging to different blocks are all 1 through successive three blocks, the binary data in the first two blocks are converted in accordance with the rule shown in Figure 3, and the binary data in the last block are converted in accordance with the rule shown in Figure 2.

With the step of conversion, the minimum and the maximum number of zeros between adjacent two ones in successive 3 bit data obtained by the conversion is, respectively, 1 and 7; therefore self-synchronization becomes possibles. Furthermore, the $T_{min}$ and the $T_w$ thereof are sufficiently large values of

$$\frac{4}{3}T$$

and

$$\frac{2}{3}T,$$

respectively, and the value of $T_{max}$ is

$$\frac{16}{3}T.$$

In addition, since the conversion step according to Figure 2 and Figure 3 has a relatively small unit of conversion and has a desirable arrangement of conversion, an actual circuit for effecting the conversion or the reverse conversion thereof can be simplified.

Subsequently, a first 3 bit binary code in every six blocks shown in Figure 1B is converted to a 4 bit binary code shown in Figure 1C in accordance with either one of bit patterns $P_1$ and $P_2$ shown in the truth table of Figure 4.

Figure 1D shows an example of a recording signal which is derived from the binary data shown in Figure

1C, in a manner that one in the binary data represents an inversion, and zero in the binary data represents a non-inversion.

The bit pattern, i.e., either one of $P_1$ and $P_2$ to be used in the above-described conversion is so determined that the magnitude of a dc or a low frequency component of a recording signal constituted by 6 blocks, such as the signal shown in Figure 1D, would be the minimum.

The calculation of the dc component is, for example, executed by an accumulation of +1 and −1 values which respectively correspond to the high level and the low level of the recording signal, through 6 blocks in which the conversion is to be effected.

In the case of the example of data shown in Figure 1B, the cumulative values of the dc component of two recording signals, from a point A through a point B including six blocks, which would be obtained by the conversion according to the bit patterns $P_1$ (1000) and $P_2$ (1010), are respectively −1 and +5. Therefore, the bit pattern $P_1$ (1000) is selected, as shown in the first block of Figure 1C. The determination of bit pattern in the seventh block is executed in the same manner as described above.

In addition, the selection of the three bit patterns, "0000"', "0001", and "1000" is permitted only when the value of $T_{max}$ would not be greater than

$$\frac{16}{3}T,$$

i.e., when the number of adjacent "0"s is smaller than eight.

Furthermore, a number L of the blocks in which the second conversion of the binary codes, i.e., the conversion according to the truth table of Figure 4, is executed, is selected in view of an upper limited frequency of low frequency component which should be eliminated. Specifically, if L is small, the upper limited frequency will be high, and on the other hand, if L is large, the upper limited frequency will be low. Furthermore, $T_{max}$ and $T_w$ are short when L is small, and are long when L is large.

In the case of the embodiment described above, the number L is selected to be six.

The values of $T_{min}$, $T_{max}$ and $T_w$ in the embodiment are calculated in terms of the number L, as follows:

$$T_{min} = \frac{3L}{3L+1} \cdot \frac{4}{3}T$$

$$T_{max} = \frac{3L}{3L+1} \cdot \frac{16}{3}T$$

$$T_w = \frac{3L}{3L+1} \cdot \frac{2}{3}T$$

Furthermore, the rule of conversion shown in Figure 4 can be replaced by a rule shown in Figure 5. In that case, the dc or the low frequency component of the recording signal is effectively reduced. On the other hand, $T_{min}$ and $T_w$ are shorter than those of the previous case.

Turning to Figures 6A and 6B, an example of a circuit, designed for conversion according to the present invention will be explained. Figure 6A shows a circuit part for executing the conversion according to the rules shown in Figures 2 and 3. Figure 6B shows a circuit part for executing the conversion according to the rule shown in Figure 4.

In Figure 6A, a plurality of binary codes which form a data train are in turn applied to a four bit shift register 1 in accordance with a clok pulse a. When the binary codes of two blocks, i.e., four bits, have been applied, the shift register 1 produces four output signals $d_{11}$, $d_{12}$ $d_{21}$, and $d_{22}$, in which $d_{12}$ are applied to an AND gate 2. If $d_{12} = d_{21} = 1$, the AND gate 2 produces an output signal of logic 1 which is applied to a selector circuit 3.

The selector circuit 3, also receiving the output signals $d_{22}$ and $d_{12}$ of the shift register 1, transmits the signal $d_{12}$ to a NOR gate 4 and to one of several parallel inputs of a three bit shift register 5 as a signal $C_{13}$ upon reception of the output signal of logic 1 of the AND gate 2. When the putput signal of the AND gate 2 takes logic 0 the selector circuit 3 passes therethrough the signal $d_{12}$. The output signal $d_{11}$ of the four bit shift register 1 is applied to the NOR gate 4 and to one of the parallel inputs of the three bit shift register 5, as a signal $C_{11}$. The output signal of the NOR gate 4 is also applied to one of the parallel inputs of the three bit shift register 5 as a signal $C_{12}$.

A divider circuit 6 receives the output signal of the AND gate 2 and the clock pulse a and produces an output clock signal which is obtained by dividing the clock pulse a by 4 when the output signal of the AND gate 2 is 1, and produces an output clock signal by dividing the clock pulse a by 2 when the output signal of the AND gate is logic 0.

The output clock signal of the divider circuit 6 is applied to a load signal input terminal of shift register 5. A serial input terminal of the shift registers is applied with a 0 level, and a shift clock input termenal thereof is

applied with a clock pulse b which has a repitition rate of 1.5 times of that of the clock pulse a.

Therefore, the data loading of the shift register 5 is effected after completion of a four bit shift of the contents of the shift register 1 when the output signal of the AND gate 2 is logic 1. While the four bit shift of the contents of the shift register 1 takes place the contents of the shift register 5 are shifted by 6 bit, and three zeros which have been applied to the serial input terminal thereof subsequent to the application of $C_{11}$, $C_{12}$, and $C_{13}$, are outputted from as the serial output terminal of the register 5, as the signals $C_{21}$, $C_{22}$, and $C_{23}$. Thus, the conversion according to the rule shown in Figure 3 is executed.

If the output signal of the AND gate 2 is 0, the data load of the shift register 5 from the parallel input terminals thereof takes place when the contents of the shift register 1 is shifted by two bits, and the signals $C_{11}$, $C_{12}$, and $C_{13}$ are outputted from the serial output terminal of the shift register 7. Thus, the conversion according to the rule shown in Figure 2 is executed.

Turning to Figure 6B, the circuit for executing the conversion shown in Figure 4 will be explained.

The output signal C of the shift register 5 in Figure 6A is applied to a serial input terminal of a six bit shift register 7. A clock input terminal of the shift register is applied with a clock pulse d which is synchronized with the clock pulse b.

A circuit (not shown) is provided for stopping the clock pulses a and b during one clock period of the clock pulse d when the last bit of a particular block, to which the conversion of Figure 4 is applied, the plurality of blocks for each conversion, i.e., six blocks in this case, is outputed from the shift register 5. With the operation of this circuit, the shift register 7 receives the signals $C_{i1}$, $C_{i2}$, $C_{i3}$ and $C_{i3}$ ($i = 1$ or 2) when the signals $C_{i1}$, $C_{i2}$ and $C_{i3}$ are produced by the shift register 5.

An output signal of the shift register 7 is applied to a four bit shift register 8 and then to a sixteen bit shift register 9 and a four bit shift register 10.

Parallel output signals of the shift register 7, which correspond to five bits from a bit which is lastly applied to the shift register 7, parallel output signals of the shift register 8, and parallel output signals of the shift register 8, and parallel output signals of the shift register 9 which correspond to four bits from a bit which is lastly applied to the shift register 9, are applied to a programme logic alley (PLA) 11.

When three bits $C_{i1}$, $C_{i2}$, and $C_{i3}$ of four bits in a block which is to be converted, are received by the shift register 8, the PLA effects its conversion operation in view of those three bits $C_{i1}$, $C_{i2}$ and $C_{i3}$ and another five bits of both sides of the block. In other words, when ($C_{i1}$, $C_{i2}$, $C_{i3}$) is (010), (001), or (100), the PLA 11 determines whether or not the bit pattern $P_2$ satisfies the condition of $T_{max}$ in view of five bits of both sides of the block.

If the bit pattern $P_2$ satisfies the condition of $T_{max}$, bit patterns $P_1$ and $P_2$ are loaded in the shift register 8 and in a four bit shift register 12 by respectively by the PLA 11. In addition, the shift operations in the shift registers 7, 8, 9, 10 and 12 are synchronized with the clock pulse d. Further, the shift registers 8 and 12 load the data from parallel input terminals thereof in synchronization with the clock pulse d, when a load signal e is applied thereto. Similarly, the shift register 10 loads the data from parallel input terminals thereof in synchronization with the clock pulse d, when a load signal f is applied thereto. In those shift registers 8, 10, and 12, central two terminals of four parallel input terminals thereof are connected to the output terminals of PLA 11, and the other two terminals in both sides are connected to receive data applied to a serial input terminal thereof.

As shown in Figure 4, the two bits $b_1$ and $b_4$ of the four bit data, which are common in both of bit patterns $P_1$ and $P_2$, are equal to $C_{i1}$ and $C_{i3}$, respectively. Since $C_{i1}$ is present at the third bit terminals of shift registers 8 and 12, and $C_{i3}$ is present at the sixth bit terminal of the shift register 7; the PLA 11 produces only the other two bits $b_2$ and $b_3$ which are not always common bit patterns $P_1$ and $P_2$, and loads them into the shift registers 8 and 12. These two bits $b_2$ and $b_3$ are also stored in a two bit latch 13.

A pair of flip-flops 14 and 16 and a pair of up-down counters 15 and 17 are provided to calculate a pair of absolute values $D_1$ and $D_2$ of the dc components from a time of loading of data for the conversion, to a time just before the next loading, which respectively represent the values when the bit pattern $P_1$ or $P_2$ is adopted.

The flip-flops 14 and 16 receive an output signal of the four bit shift registers 8 and 12, respectively, and apply an output signal to an up-down control input terminal of the up-down counters 15 and 17. The flip-flops 14 and 16 are inverted in state in response to logic "1" signls from the shift registers 8 and 12, respectively. The up-down counters 15 and 17 count up the clock pulse d when a high level signal from the flip-flops 14 and 16 is applied thereto, and count down the same when a low level signal from the flip-flops 14 and 16 is applied thereto. Thus, the up-down counters 15 and 17 accumulate the dc component $D_1$ and $D_2$, respectively. In addition. the up-down counters 15 and 17 are initially set at an absolute value of dc component $D_1$ and $D_2$ of previous data blocks, and it is assumed that $D_1$ is equal to $D_2$ ($D_1 = D_2$).

The accumulated values of dc components $D_1$ and $D_2$ are compared in a comparator 18.

These calculation and comparison processes have been completed when the three bits in the bit pattern $P_1$ is received in the shift register 10. If $D_1$ is larger than $D_2$ ($D_1 > D_2$), the comparator 18 produces a load signal f which is applied to a load signal input terminal of the shift register 10. Then, the signals $b_2$ and $b_3$ of the bit pattern $P_2$, which are stored in the two bit latch 13, are loaded to the shift register 10 at the timing of the next clock pulse signal d.

Conversely, if $D_2$ is equal to or smaller than $D_2$ ($D_1 \leqq D_2$), the load signal f is not produced, and the contents of the shift register 10 remains as the bit pattern $P_1$.

The comparator 18 also produces either one of a pair of control signals g and g' which are respectively

connected to the flip-flop 14 and the up-down counter 15, and the flip-flop 16 and the up-down counter 17 which corresponds to the bit pattern of the larger dc component. Subsequently, the state of the flip-flop (14 or 16) and the up-down counter (15 or 17) which are applied with the control signal (g or g') is set to the same as that of others which correspond to the bit pattern having a lower dc component. Thus, the circuits are set to the initial state of the calculation of the dc compnents $D_1$ and $D_2$.

When the three bit data ($C_{i1}$, $C_{i2}$, $C_{i3}$) is equal to (0,0,0) or (1,0,1), and when the bit pattern $P_2$ does not satisfy the condition of $T_{max}$, the bit pattern $P_1$ is loaded in both of the shift registers 8 and 12. In that case, the load signal f is not produced since the dc components $D_1$ and $D_2$ are always equal to each other ($D_1 = D_2$).

Turning to Figure 7, a demodulator circuit for reversing the data converted by the circuit shown in Figures 6A and 6B will be explained.

As shown in Figure 7, a two bit shift register 20, a one bit shift register 21, and a four bit shift register 22 are connected in series relation to form a seven bit shift register.

Converted input data is applied to the two bit shift register 20 in accordance with a clock pulse h. Each binary code of the input signals is applied to the seven bit shift register in two different manners, depending on the type of block to which the binary code belongs.

More specifically, if the binary code is of the block to which the conversion according to Figure 4 is not effected, the binary code is in turn received by the seven bit shift register. On the other hand, in the case where the conversion of Figure 4 is effected, a reverse process of the conversion of Figure 4 is effected when four bits of such bolck are entered to the two bit shift register 20, to the one bit shift register 21, and to a first bit of the four bit shift register 22.

The resulting three bits of the reverse conversion is maintained in a second bit of the two bit shift register 20, in the one bit shift register 21, and in the first bit of the four bit shift register 22. The first bit of the two bit shift register 21 receives a binary code of a first bit of the next block. In this state, the contents of a second, a third, and a fourth bit of the four bit shift register 22 remain unchanged.

Since the conversion of Figure 4 is expressed as $C_{i1} = b_1$, $C_{i3} = b_4$, and $C_{i2} = \overline{b_1 + b_2 + b_4}$, the reverse process will be effected by entering $C_{i2}$ in the one bit register 21 by means of a three input NOR gate 23, a control signal i, and selector 24, and by shifting the contents of the two bit shift register 20 by one bit at the same time. In this state, the clock pulse b is stopped by an AND gate 35 and the control signal i which has a 0 level only at this point, and therefore the contents of the four bit shift register 22 are kept unchanged. After the process described above, the shift registers 20, 21 and 22 contain binary codes of blocks which are uniformly formed by three bits.

A NOR gate 26, an OR gate 27, a divider circuit 29, and a four bit shift register 28 are provided to effect a process which has an inverse relation to the conversions shonw in Figure 2 and Figure 3. When $C_{21} = C_{22} = C_{23} = 0$, the NOR gate 26 produces an output signal of 1. Subsequently, two bits $d_{12}$ and $d_{21}$ to be supplied to the four bit shift register 28 are set to 1 ($d_{12} = d_{21} = 1$). The other two bits $d_{11}$ and $d_{22}$ are connected to two bits $C_{11}$ and $C_{13}$ of the four bit shift register 22, respectively. Thus, a process having an inverse relation to the conversion of Figure 3 is executed.

When the condition $C_{21} = C_{22} = C_{23} = 0$ is not satisfied, the NOR gate 26 produces an output signal of 0; therefore, the two bits $d_{11}$ and $d_{12}$ will be equal to the two bits $C_{11}$ and $C_{13}$ ($d_{11} = C_{11}$, $d_{12} = C_{13}$). Thus, a process having an inverse relation to the conversion of Figure 2 is executed.

These binary codes $d_{22}$, $d_{21}$, $d_{12}$, and $d_{11}$ are loaded to the shift register 28 and then are outputted one by one in accordance with a clock pulse j.

The divider circuit 29 for producing a load signal of the four bit shift register 28 is supplied with the output signal of the NOR gate 26 and the clock pulse j. When the output signal of the NOR gate 26 is 0, the divider circuit 29 divides the clock pulse j by two, and when the output signal of the NOR gate 26 is 1, the divider circuit 29 divides the block pulse j by four.

Therefore, when the output signal of the NOR gate 26 is 0, the next data is loaded to the four bit shift register 28 at a time when two bits of the inverse signal are outputted therefrom. On the other hand, when the output signal of the NOR gate 26 is 1, the next data is loaded to the four bit shift register 28 at a time when four bits of the inverse signal are outputted therefrom. Thus, successive processes of inverse conversion take place by every one or two blocks, depending on the type of conversion effected.

As described above, the binary data conversion method according to the present invention features a step of converting binary codes in each M bit block of an input data train to N bit binary codes, and a step of converting the N bit binary codes to J bit binary codes at every L blocks of the data train.

Therefore, the converting method can be designed to have a sufficient length of at least one of $T_{min}$ and $T_w$ and to eliminate the dc or low frequency component. Furthermore, the relation between the lengths of $T_{min}$ and $T_w$ and the frequency of the component to be eliminated, can be determined by selecting the values of L and J described above.

In addition, the converting method according to the present invention has an advantage that the construction of the demodulator circuit is relatively simple.

As an example, when the above described values M and N are selected to be 2 and 3 (M=2, N=3) and conversions shown Figures 2 and 3 are adopted as the case of the described embodiment, sufficiently long values of $T_{min}$ and $T_w$ are obtained. The specific values of $T_{min}$, $T_{max}$ and $T_w$ are as follows:

$$T_{min} = \frac{3L}{3L+1} \cdot \frac{4}{3}T \cdot T_{max} = \frac{3L}{3L+1} \cdot \frac{16}{3}T$$

$$T_w = \frac{3L}{3L+1} \cdot \frac{2}{3}T.$$

Moreover, by appropriate design of each conversion, the construction of the demodulator circuit can be further simplified.

In the above equations, when the value of L is 6, as the case of the embodiment, the values of $T_{min}$ and $T_w$ are as follows:

$$T_{min} = 1.26T, T_w = 0.63T$$

In this case $T_{max}$ is 11% less than EFM, and $T_w$ is 34% greater than EFM.

Since the value of $T_w$, as well as the value of $T_{min}$, contributes to the amplitude of the signal at a detection point in the case of playback mode of the record and playback system employing a laser beam, the converting method according to the present invention has greater amplitude at the detection point, under the range of practical density of recording, that that of the EFM technique.

Thus, the converting method according to the present invention has greater tolerances against noise or fluctuation of the time axis (jitter) of the playback signal.

Due to the advantages described above, the converting method can be applied to various systems of data recording or data transmission, such as a playback system using variation of the electrostatic capacity or a magnetic recording and playback system, to perform high-density data recording or high density data transmission.

Above, a preferred embodiment of the present invention has been described. It should be understood, however, that the foregoing description is for illustrative purpose only, and is not intended to limit the scope of the invention. Rather, there are numerous equivalents to the preferred embodiment, and such are intended to be covered by the appended claims.

CLAIMS

1. A method for converting a binary data train comprising:

a first step for dividing said binary data train into a plurality of successive blocks, each having M (M being a natural number) bit data;

a second step for converting each of said M bit data into N (N being a natural number, and where $N \geqq M + 1$) bit data; and

a third step for converting an N bit data in every L (L being a natural number) number of said blocks into J (J being a natural number) bit data.

2. A method as recited in claim 1, wherein J is equal to or greater than N+1, wherein said N bit data comprises a first type corresponding to one J bit data and a second type corresponding to at least two J bit data, and wherein said third step comprises a step for selecting one of said at least two J bit data, so as to minimize a dc and a low frequency component of a signal to be obtained after a conversion of the binary data train.

3. A method as recited in claim 1, wherein M and N are equal to 2 and 3, respectively,

wherein said 2 bit data comprises a first data, a second data, a third data, and a fourth data, and wherein said first step comprises a step for converting said first, second, third, and fourth data into binary codes of "010", "001", "100", and "101", respectively, when data of two adjacent of said blocks are not combinations of "second data·third data", "second data·fourth data", "fourth data·third data", or "fourth data·fourth data", and converting each of said combinations into binary codes of "010·000", "001·000", "100·000", and "101·000", respectively, when said data of two adjacent blocks is any one of said combinations.

4. A method as recited in claim 3, wherein said first, second, third, and fourth data are binary codes "00", "01", "10", and "11", respectively.

5. A method as recited in claim 2, wherein N and J are equal to 3 and 4, respectively, and wherein said third step comprises a step for converting 3 bit data "010", "001", "000", and "101" into 4 bit data of either one of "0010" and "0000", eigher one of "0101" and "0001", either one of "1010" and "1000", "0100", and "1001", respectively.

6. A method as recited in claim 2, wherein N and J are equal to 3 and 5, respectively, and wherein said third step comprises a step for converting 3 bit data "010", "001", "100", "000", and "101" into 5 bit data "01010", either one of "00010" and "01000", "00101", either one of "01001" and "00001", "10100", either one of "10010" and "10000", either one of "00100" and "00000", and either one of "10101" and "10001", respectively.

7. A method for converting a binary data train comprising:

a first step for dividing said binary data train into a plurality of successive blocks each having 2 bit data of one of first, second, third, and fourth data;

a second step for converting said first, second, third, and fourth data into three bit binary codes of "010", "001", "100" and "101", respectively, when data of two adjacent of said blocks are not combinations of "second data third data", "second data·fourth data", "fourth data·third data", or "fourth data·fourth data", and converting each of said combinations into binary codes of "010·000", "001·000", "100·000" and "101·000", respectively, when said data of two adjacent of blocks is any one of said combinations.

8. A method as recited in claim 7, wherein said first, second, third, and fourth data are binary codes "00", "01", "10", and "11", respectively.

9. A method of converting a binary data train substantially as hereinbefore described with reference to the accompanying drawings.

Printed for Her Majesty's Stationery Office, by Croydon Printing Company Limited, Croydon, Surrey, 1983.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.